Class 7: Machine Learning 1

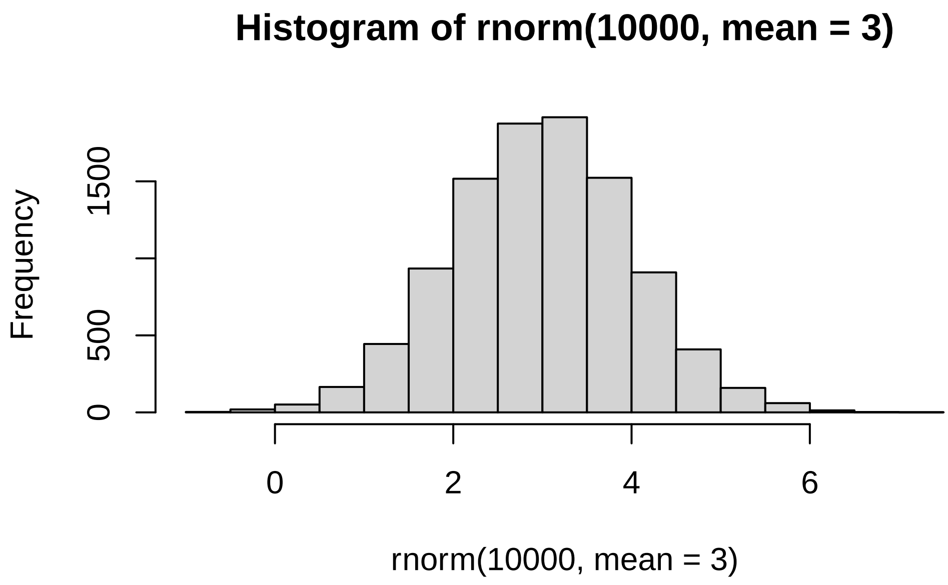
SungWoo Park (PID: 69026846)

Clustering

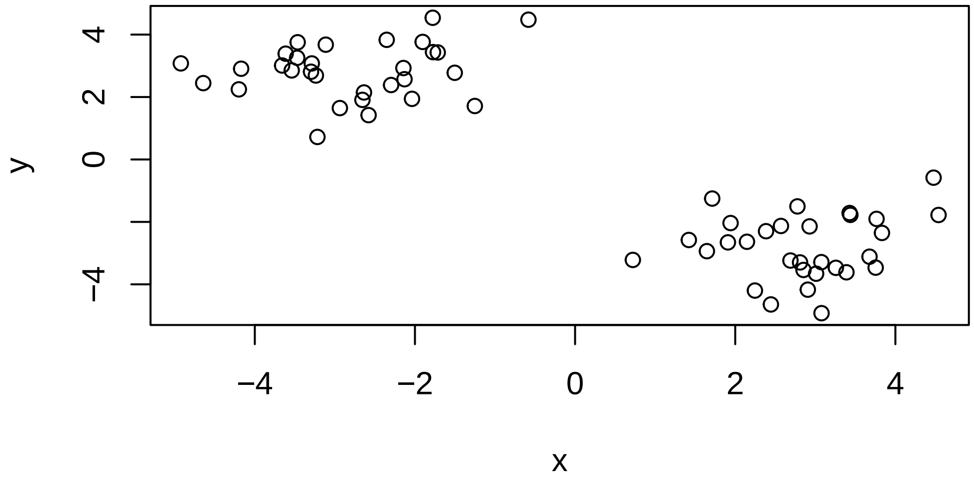
We will start with k-means clustering, one of the most prevalent of all clustering methods.

To get started let's make some data up:

```r
hist ( rnorm(10000, mean=3) )
```



```r
tmp <- c( rnorm(30, 3), rnorm(30, -3) )
x <- cbind(x=tmp, y=rev(tmp) )
plot(x)
```



The main function in R for K-means clustering is called 'kmeans()'.

```
k <- kmeans(x, centers=2, nstart=20)
k
```

```
K-means clustering with 2 clusters of sizes 30, 30

Cluster means:
          x         y
1  2.827604 -2.804050
2 -2.804050  2.827604

Clustering vector:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 53.081 53.081
 (between_SS / total_SS =  90.0 %)

Available components:
```

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"      
```

> Q1. How many points are in each cluster

```
k$size
```

```
[1] 30 30
```

> Q2. The clustering result i.e. membership vector?

```
k$cluster
```

```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

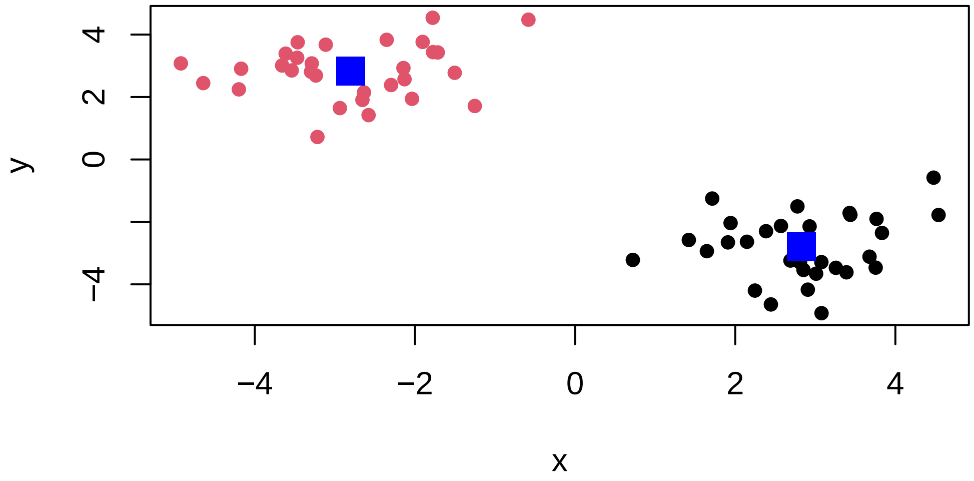
> Q3. Cluster centers

```
k$centers
```

```
          x         y
1  2.827604 -2.804050
2 -2.804050  2.827604
```

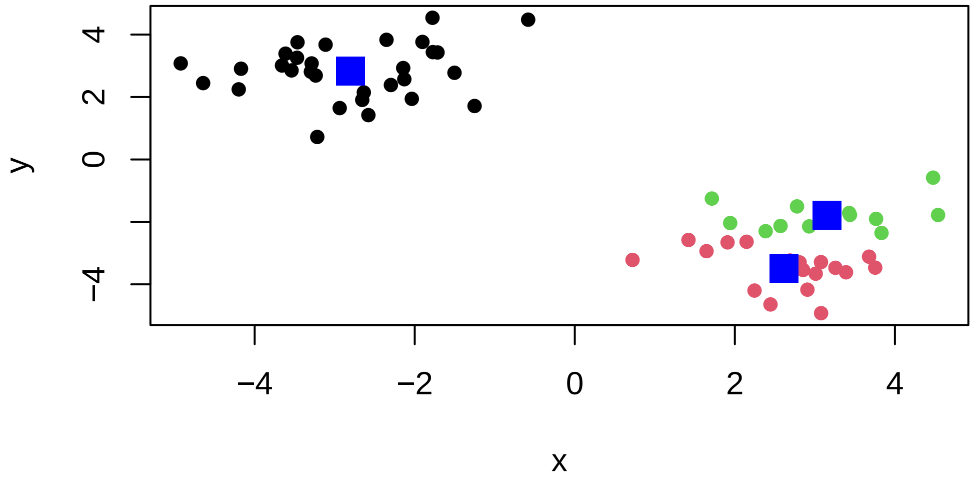
> Q4. Make a plot of out data colored by clustering results with optionally the cluster centers shown.

```
plot(x, col=k$cluster, pch=16 )
points(k$centers, col="blue", pch=15, cex=2)
```



Q5. Run kmeans again but cluster into 3 groups and plot the results like we did above.

```
k3 <- kmeans(x, centers=3, nstart=20)
plot(x, col=k3$cluster, pch=16 )
points(k3$centers, col="blue", pch=15, cex=2)
```



Hierarchial

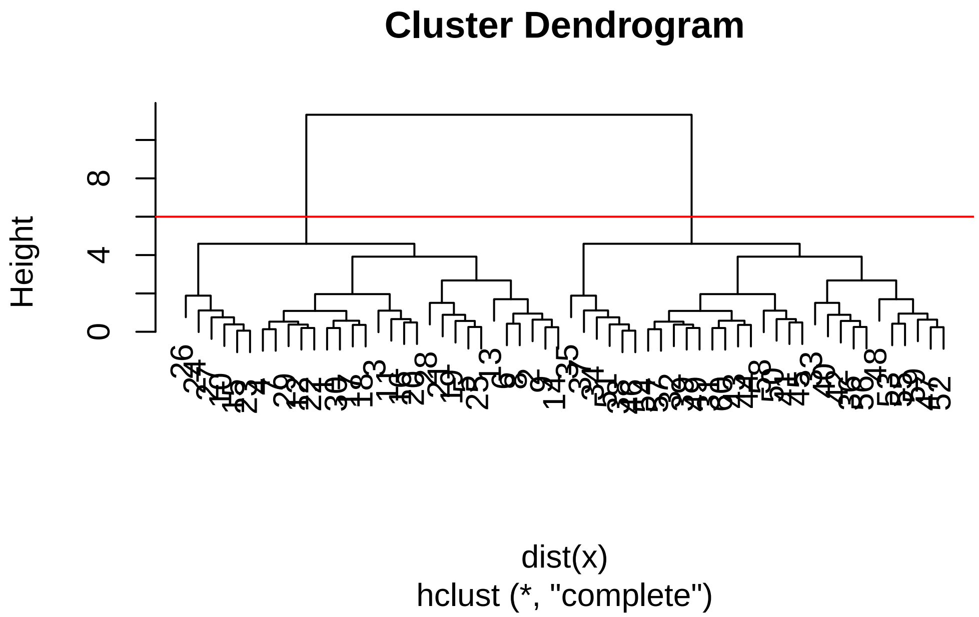
First we need to calculate point (dis)similarity as the Euclidean distance between observations dist_matrix <- dist(x) The hclust() function returns a hierarchical clustering model hc <- hclust(d = dist_matrix) the print method is not so useful here hc

```
hc <- hclust( dist(x) )
hc
```

```
Call:
hclust(d = dist(x))

Cluster method   : complete 
Distance         : euclidean 
Number of objects: 60 
```

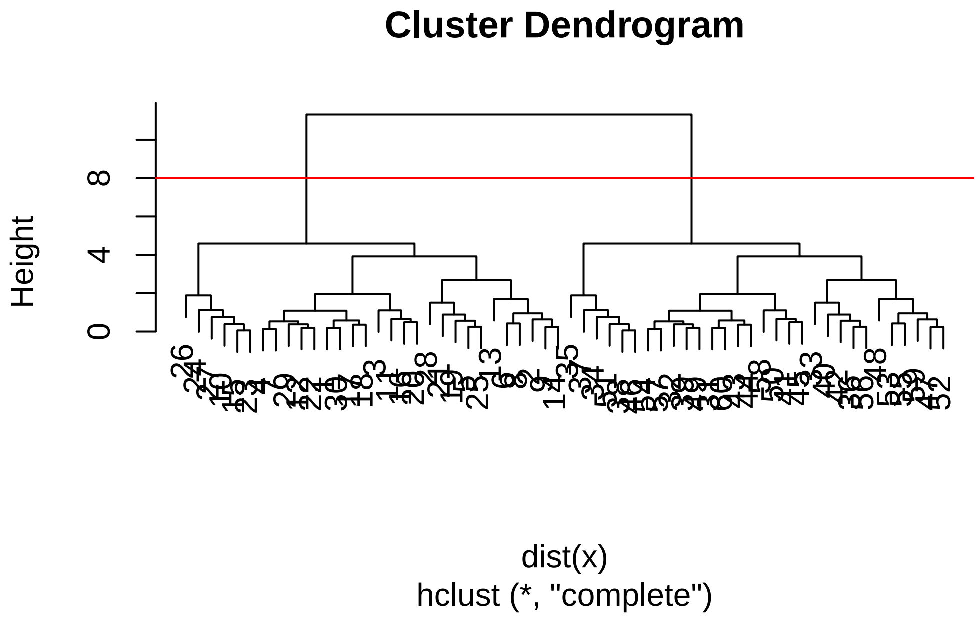
```
plot(hc)
abline(h=6, col="red")
```



Draws a dendrogram

The function to get our clusters/groups from a hclust object is called ‘cutree()’

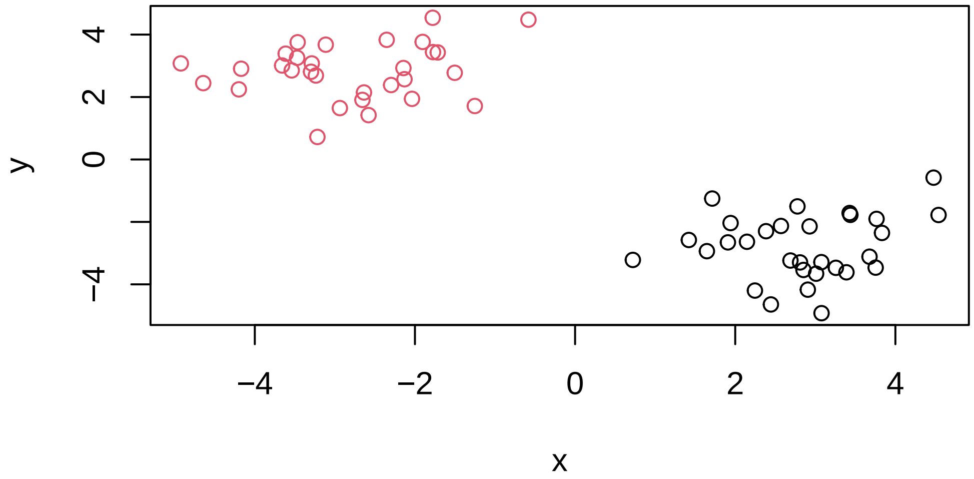
```
plot(hc)
abline(h=8, col="red")
```



```r
grps <- cutree(hc, k=2)
```

Q. Plot our hclust results in terms of our data colored by cluster membership.

```r
plot(x, col=grps)
```



Principal Component Analysis (PCA)

Principal components are new low dimensional axis closest to the observations. The data have maximum variance along PC1 which makes the first few PCs useful for visualizing our data and as a basis for further analysis.

> Q1. How many rows and columns are in your new data frame named x? What R functions could you use to answer this questions? 17 rows and 5 column. Use dim(x), nrow(x), ncol

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url)
head(x)
```

```
               X England Wales Scotland N.Ireland
1         Cheese     105   103      103        66
2  Carcass_meat      245   227      242       267
3    Other_meat      685   803      750       586
4           Fish     147   160      122        93
5 Fats_and_oils      193   235      184       209
6         Sugars     156   175      147       139
```

```
dim(x)
```

```
[1] 17  5
```

```
rownames(x) <- x[,1]
x <- x[,-1]
head(x)
```

```
               England Wales Scotland N.Ireland
Cheese             105   103      103        66
Carcass_meat       245   227      242       267
Other_meat         685   803      750       586
Fish               147   160      122        93
Fats_and_oils      193   235      184       209
Sugars             156   175      147       139
```

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url, row.names=1)
head(x)
```

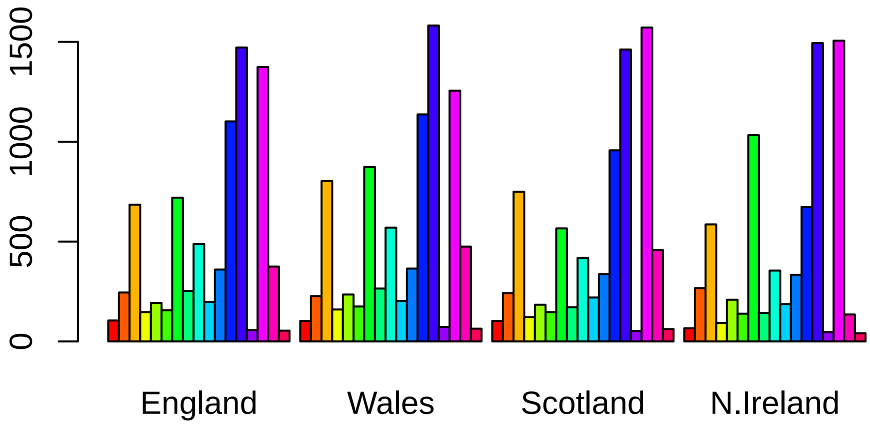
```
               England Wales Scotland N.Ireland
Cheese             105   103      103        66
Carcass_meat       245   227      242       267
Other_meat         685   803      750       586
Fish               147   160      122        93
Fats_and_oils      193   235      184       209
Sugars             156   175      147       139
```

> Q2. Which approach to solving the 'row-names problem' mentioned above do you prefer and why? Is one approach more robust than another under certain circumstances?

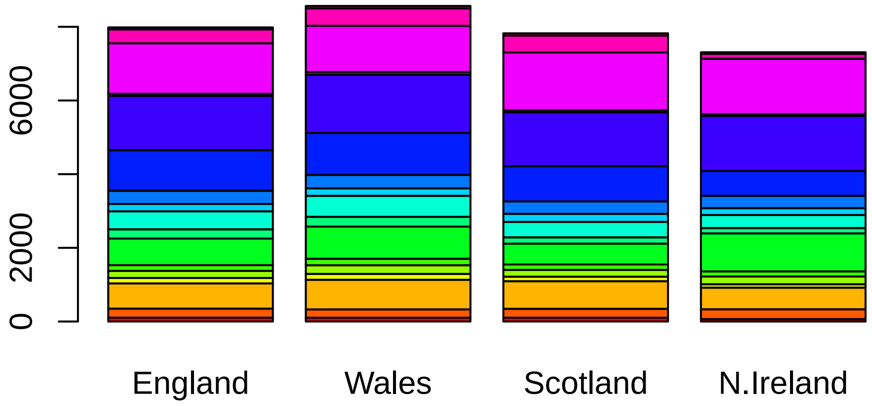
I prefer changing the row.names=1. This way is more robust than set the rowname and delete one.

> Q3: Changing what optional argument in the above barplot() function results in the following plot?

```
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```



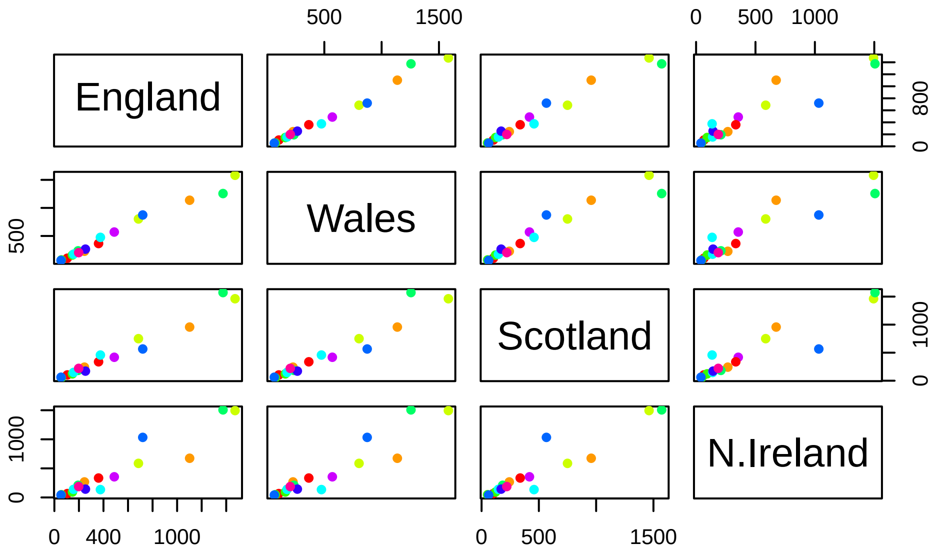
```
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```



Q5: Generating all pairwise plots may help somewhat. Can you make sense of the following code and resulting figure? What does it mean if a given point lies on the diagonal for a given plot?

If a given point lies on the diagnol means that the measurement is similar between two countries. The more a given point lies out of the diagnol means that the measurement is different.

```
pairs(x, col=rainbow(10), pch=16)
```



The main function for PCA in base R is called ‘prcomp()’

It wants the transpose (with the ‘t()’) of our food data for analysis.

> Q6. What is the main differences between N. Ireland and the other countries of the UK in terms of this data-set?

The correlation of northern Ireland with other countries are not as linear as other countries

```
t(x)
```

```
          Cheese Carcass_meat  Other_meat  Fish Fats_and_oils  Sugars
England      105           245         685   147            193     156
Wales        103           227         803   160            235     175
Scotland     103           242         750   122            184     147
N.Ireland     66           267         586    93            209     139
          Fresh_potatoes  Fresh_Veg  Other_Veg  Processed_potatoes
England              720        253        488                 198
Wales                874        265        570                 203
Scotland             566        171        418                 220
N.Ireland           1033        143        355                 187
          Processed_Veg  Fresh_fruit  Cereals  Beverages Soft_drinks
England             360         1102     1472         57        1374
```

```
Wales                 365          1137      1582        73           1256
Scotland              337           957      1462        53           1572
N.Ireland             334           674      1494        47           1506
          Alcoholic_drinks  Confectionery
England                  375            54
Wales                    475            64
Scotland                 458            62
N.Ireland                135            41
```

```
pca <- prcomp( t(x) )
pca$x
```

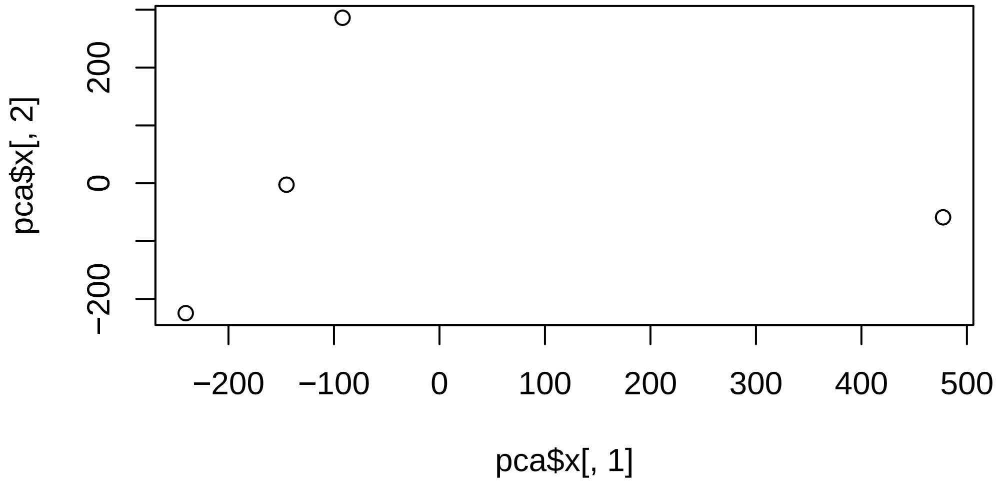
```
                PC1         PC2        PC3           PC4
England   -144.99315   -2.532999 105.768945 -9.152022e-15
Wales     -240.52915 -224.646925 -56.475555  5.560040e-13
Scotland   -91.86934  286.081786 -44.415495 -6.638419e-13
N.Ireland  477.39164  -58.901862  -4.877895  1.329771e-13
```

```
summary(pca)
```

```
Importance of components:
                            PC1      PC2      PC3       PC4
Standard deviation     324.1502 212.7478 73.87622 2.921e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

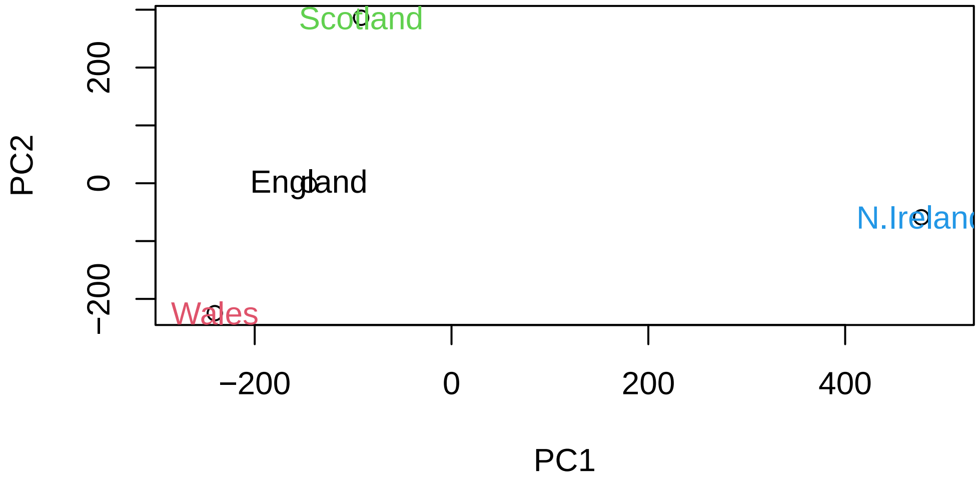
One of the main results that look for is called the “score plot” a.k.a. PC plot, PC1 vs PC2 plot… > Q7. Complete the code below to generate a plot of PC1 vs PC2. The second line adds text labels over the data points.

```
plot( pca$x[,1], pca$x[,2])
```



Q8. Customize your plot so that the colors of the country names match the colors in our UK and Ireland map and table at start of this document.

```
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2", xlim=c(-270,500))
text(pca$x[,1], pca$x[,2], colnames(x), col=c(1,2,3,4))
```



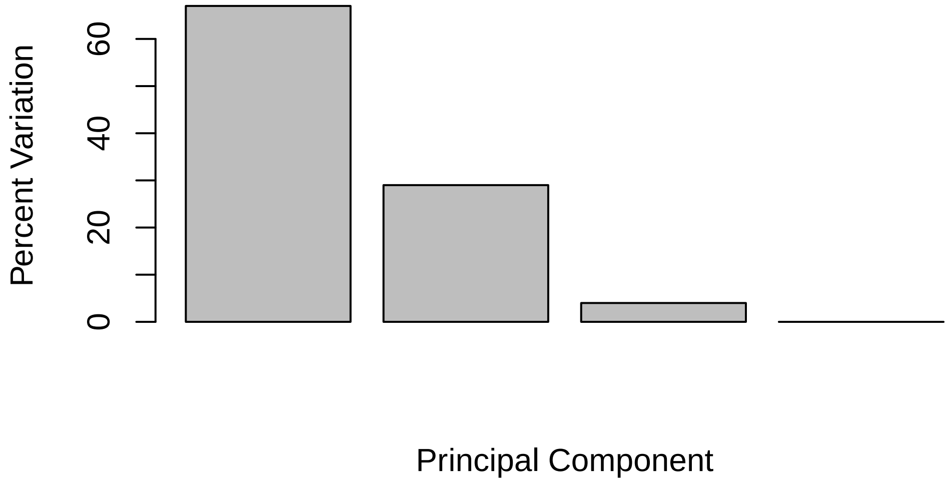
```
v <- round( pca$sdev^2/sum(pca$sdev^2) * 100 )
v
```

```
[1] 67 29  4  0
```

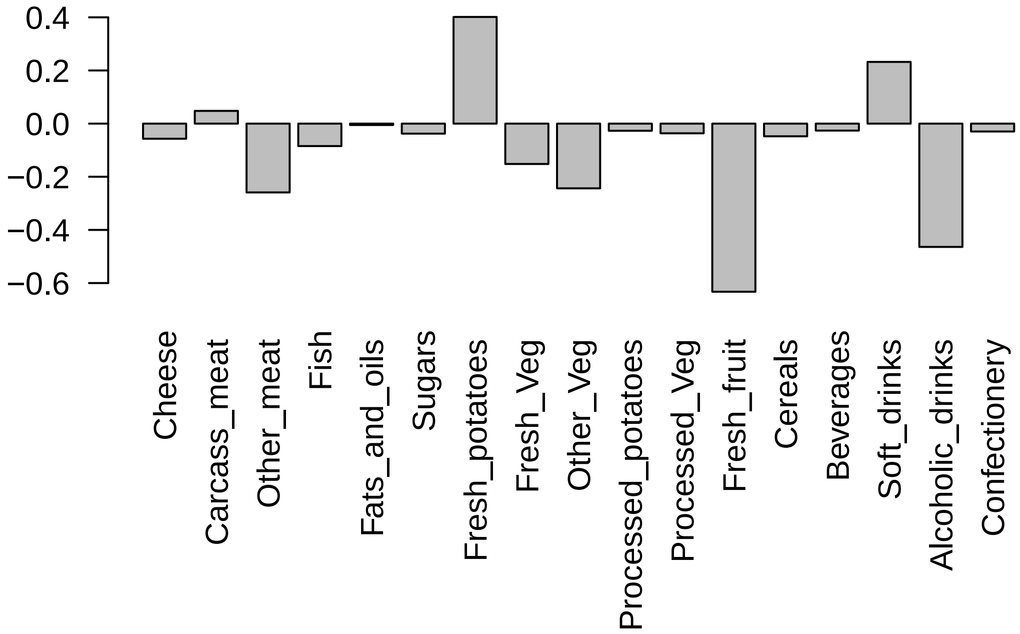
```
## or the second row here...
z <- summary(pca)
z$importance
```

```
                             PC1       PC2      PC3          PC4
Standard deviation     324.15019 212.74780 73.87622 2.921348e-14
Proportion of Variance   0.67444   0.29052  0.03503 0.000000e+00
Cumulative Proportion    0.67444   0.96497  1.00000 1.000000e+00
```

```
barplot(v, xlab="Principal Component", ylab="Percent Variation")
```



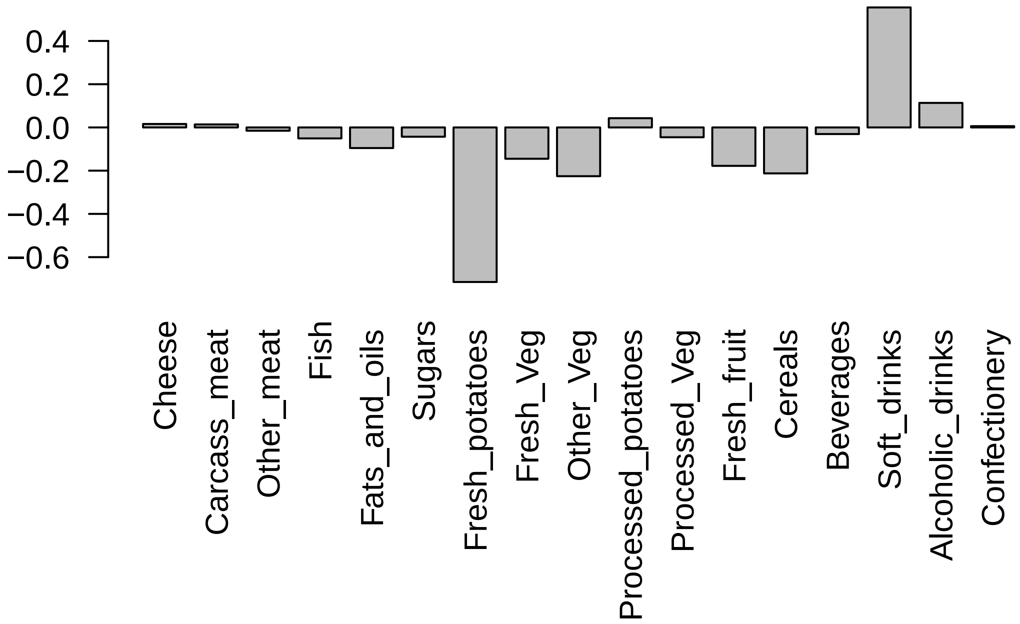
```
## Lets focus on PC1 as it accounts for > 90% of variance
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )
```



> Q9: Generate a similar 'loadings plot' for PC2. What two food groups feature prominantely and what does PC2 maninly tell us about?

Fresh_potatoes and Soft_drinks. Soft drinks account for scotland and Wales account for Fresh_potatoes.

```
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,2], las=2 )
```


0.4
0.2
0.0
−0.2
−0.4
−0.6
Cheese
Carcass_meat
Other_meat
Fish
Fats_and_oils
Sugars
Fresh_potatoes
Fresh_Veg
Other_Veg
Processed_potatoes
Processed_Veg
Fresh_fruit
Cereals
Beverages
Soft_drinks
Alcoholic_drinks
Confectionery